# 가우시안 정규기저를 이용한 $GF(2^m)$상의 워드-레벨 곱셈기

정회원 김 창 훈*, 준회원 권 윤 기**, 김 태 호*, 정회원 권 순 학**, 홍 춘 표*

# Word Level Multiplier for $GF(2^m)$ Using Gaussian Normal Basis

Chang Hoon Kim* ***Regular Member***, Yun Ki Kwon**, Tae Ho Kim* ***Associate Members***, Soonhak Kwon, Chun Pyo Hong ***Regular Members***

요 약

본 논문에서는 타원곡선 암호 시스템(Elliptic Curve Cryptosystem: ECC)을 위한 $GF(2^m)$상의 새로운 워드-레벨 곱셈기를 제안한다. 제안한 곱셈기는 원소표기법으로 가우시안 정규기저(Gaussian Normal Basis: GNB)를 이용하며, $\lceil m/w \rceil$ 클럭 사이클 마다 곱셈 연산의 결과를 출력한다. 여기서 $w$는 워드크기이다. 제안한 워드-레벨 곱셈기를 Xilinx XC2V1000 FPGA칩을 이용하여 구현한 후 기존에 제안된 워드-레벨 곱셈기와 성능을 비교 분석한 결과, 가장 낮은 최대 처리기 지연시간(critical path delay)을 가진다.

**Key Words** : Finite Field, $GF(2^m)$ Multiplication, Word Level Multiplier, Gaussian Normal Basis, VLSI

ABSTRACT

This paper presents a new word level multiplier over $GF(2^m)$ for elliptic curve cryptosystem. The proposed multiplier uses Gaussian normal basis representation and produces multiplication results at a rate of one per $\lceil m/w \rceil$ clock cycles, where $w$ is the selected word size. We implement the proposed design using Xilinx XC2V1000 FPGA device. Our design has significantly less critical path delay compared with previously proposed hardware implementations.

## I. 서 론

최근 유한체 연산은 오류제어 코드와 암호 시스템 등 다양한 분야에 응용되고 있다[1]-[3]. 이러한 응용에 사용되는 유한체는 $GF(p)$, $GF(2^m)$이 있으며 여기서 $p$는 소수이다. 특히 $GF(2^m)$은 $GF(2)$의 $m$차원 확장 벡터로서 산술 연산에 있어 캐리가 발생하지 않기 때문에 하드웨어 구현에 적합하다. $GF(2^m)$상의 중요한 연산으로는 덧셈, 곱셈, 지수, 역원 연산이 있으며 덧셈 연산은 비트별 XOR 연산으로 쉽게 구현이 가능하지만 나머지 연산들은 매우 복잡하다. 특히 $GF(2^m)$상의 역원 연산은 면적 및 속도에 있어 가장 복잡한 연산이며 반복적인 곱셈 연산으로 이루어진다. 따라서 효율적인 곱셈기 구현은 필요하다.

$GF(2^m)$상의 산술 연산에 있어 덧셈과 뺄셈을 제외한 나머지 연산들은 기저의 선택에 따라 그 성능이 좌우된다. 대표적인 $GF(2^m)$의 원소표기법으로

※ 이 논문은 2005년도 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2005-041-D00763).
* 대구대학교 정보통신공학과(chkim@dsp.daegu.ac.kr, thkim@dsp.daegu.ac.kr, cphong@daegu.ac.kr)
** 성균관대학교 수학과(drmath@skku.edu, shkwon@skku.edu)
논문번호 : KICS2006-08-336, 접수일자 : 2006년 8월 3일, 최종논문접수일자 : 2006년 2월 10일

다항식 기저(Polynomial Basis: PB)와 정규기저(Normal Basis: NB)가 있다. 각 원소표기법은 장점과 단점을 가지는데, NB를 이용할 경우 $A^{2^s}$연산을 간단한 $s$-비트 순환 쉬프트 연산으로 구현할 수 있는 반면 곱셈 연산이 매우 복잡하며 하드웨어 구조가 규칙적이지 못하다. PB를 이용한 산술 연산기(Arithmetic Unit: AU)의 하드웨어 구현은 서로 다른 $m$에 대해 높은 규칙성 및 확장성을 가진다. 이러한 이유로 NB 보다 PB를 이용한 $GF(2^m)$상의 AU에 관한 연구결과가 더 많이 발표되었다[4],[5].

GNB는 NB의 특별한 경우로서 PB와 함께 IEEE 1363[1], NIST[2] 등의 다양한 표준으로 채택되었으며, 8로 나누어떨어지지 않는 모든 양의 정수 $m$에 대하여 존재한다[1]. GNB는 정수 $k$에 의해 결정되며, GNB 타입 $k$라 부른다. 여기서 GNB 곱셈기의 속도 및 하드웨어 복잡도는 $k$에 의해 결정된다. Kwon 등[6]은 $GF(2^m)$상에서 GNB를 이용한 효율적인 비트-레벨 곱셈 알고리즘을 유도하였으며, GNB 타입 2, 4의 경우에 대해 VLSI 구조를 제안하였다.

$GF(2^m)$상의 곱셈기는 비트-시리얼 및 비트-패러럴 구조로 구분할 수 있는데 일반적으로 비트-패러럴 구조는 비트-시리얼 구조에 비해 데이터 처리속도가 빠르지만 하드웨어 복잡도가 높다는 단점이 있다. 이러한 면적 및 속도의 상충관계를 개선하기 위하여 워드-레벨 곱셈기가 제안되었다[7]-[9]. 워드-레벨 구조는 데이터를 일정한 크기의 워드 단위로 나눈 후, 워드 단위로 처리 및 전송하며 데이터 크기가 $m$-비트이고 워드의 크기가 $w$-비트이면 워드의 개수는 $L=\lceil m/w \rceil$이 된다. 비트-시리얼 구조는 $m$ 클럭 사이클마다 결과를 출력하지만, 워드-레벨 구조는 $L$클럭 사이클마다 결과를 출력한다. 워드-레벨 구조는 워드의 크기가 커질수록 연산 시간을 단축할 수 있으나 하드웨어 복잡도가 증가한다. 그러나 면적 및 속도를 만족시키는 가장 적합한 워드 크기를 찾는다면 연산시간 및 하드웨어 복잡도에 있어 상충 관계를 개선할 수 있다.

본 논문에서는 ECC를 위한 $GF(2^m)$상의 새로운 워드-레벨 곱셈기를 제안한다. 제안한 곱셈기는 원소표기법으로 GNB를 이용하며, $\lceil m/w \rceil$클럭 사이클 마다 곱셈 연산의 결과를 출력한다. 여기서 $w$는 워드크기이다. 제안한 워드-레벨 곱셈기를 Xilinx XC2V1000 FPGA칩을 이용하여 구현한 후 기존에 제안된 워드-레벨 곱셈기와 성능을 비교 분석한 결과, 가장 낮은 최대 처리기 지연시간(critical path delay)을 가진다. 또한 기존의 곱셈기는 선택된 원시 기약 다항식에 따라 서로 다른 하드웨어 구조를 가지지만 본 논문에서 제안된 곱셈기는 동일한 GNB 타입만 가지면 동일한 형태의 하드웨어 구조를 얻을 수 있다. 제안한 곱셈기의 최대 처리기 지연시간 및 하드웨어 복잡도는 GNB의 타입 $k$와 $w$에 의존하기 때문에 $w$의 선택에 따라 속도 및 면적에 있어 상충관계를 개선 할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서 GNB 타입 $k$를 이용한 $GF(2^m)$상의 곱셈 알고리즘을 알아본 후 3절에서 GNB를 이용한 $GF(2^m)$상의 워드-레벨 곱셈기를 설계한다. 4절에서는 본 논문에서 제안한 워드-레벨 곱셈기의 FPGA 구현 결과와 기존에 제안된 곱셈기들의 성능을 비교 분석한다. 끝으로 5절에서 결론을 맺는다.

## Ⅱ. GNB를 타입 $k$를 이용한 $GF(2^m)$상의 곱셈 알고리즘

### 2.1 $GF(2^m)$의 GNB 타입 $k$

$m$, $k$를 소수 $p\neq 2$에 대해, $p=mk+1$인 양의 정수라 하고, $K=<\tau>$는 $GF(p)^\times$상에서 위수(order) $k$인 유일한 부분군이라 하자. $\beta$가 $GF(2^{mk})$의 $p$번째 원시근이라 하면, 아래 원소

$$\alpha=\sum_{j=0}^{k-1}\beta^{\tau^j} \qquad (1)$$

을 $GF(2)$상의 타입 $(m, k)$ 가우스 주기(gauss period)라 한다. $Ord_p 2$를 mod $p$에 대한 2의 위수라 하고, $\gcd(mk/ord_p 2,\ m)=1$이라고 가정하면, $\alpha$는 $GF(2^m)$상의 정규 원소이다. 즉, $0\le i\le m-1$에 대해, $a_i=\alpha^{2^i}$라 놓으면, $\{a_0, a_1, a_2, \cdots, a_{m-1}\}$은 $GF(2)$상의 $GF(2^m)$에 대한 기저이고, 이를 $GF(2^m)$의 GNB 타입 $k$라 부른다. $K=<\tau>$가 순환군 $GF(p)^\times$상의 위수 $k$인 부분군이기 때문에, 잉여군(quotient group) $GF(p)^\times/K$는 위수 $m$인 순환군이고, 군의 생성원은 $2K$이다. 따라서 $GF(p)^\times$의 coset decomposition을 식 (2)와 같이 disjoint union으로 나타낼 수 있다.

$$GF(p)^\times=K_0\cup K_1\cup K_2\cup\cdots\cup K_{m-1} \qquad (2)$$

여기서 $K_i=2^iK(0\le i\le m-1)$이고 $GF(p)^\times$의 모든 원소는 임의의 $0\le s\le k-1$과 $0\le t\le m-1$에 대해 $\tau^s2^t$로 유일하게 표현되고, $0\le i\le m-1$에 대해 식 (3)을 얻을 수 있다.

$$\begin{aligned}\alpha\alpha_i &= \sum_{s=0}^{k-1}\beta^{\tau^s}\sum_{t=0}^{k-1}\beta^{\tau^t 2^i} = \sum_{s=0}^{k-1}\sum_{t=0}^{k-1}\beta^{\tau^s(1+\tau^{t-s}2^i)} \\ &= \sum_{s=0}^{k-1}\sum_{t=0}^{k-1}\beta^{\tau^s(1+\tau^t 2^i)}\end{aligned} \tag{3}$$

예제로 $GF(2^7)$이 GNB 타입 4라 하자. 여기서 $m$=7, $k$=4이다. 즉, $p$=29=$mk$+1이다. 이 경우 $GF(29)^\times$상에서 위수 4인 유일한 순환군은 $K$={1,$2^7$,$2^{14}$,$2^{21}$}={1,12,28,17}이다. 여기서 $\beta$를 $GF(2^{28})$상에서 29번째 원시근이라 하고, $\tau$=12로 두면, $GF(2^7)$상의 정규 원소 $\alpha$는 $\alpha=\beta+\beta^{12}+\beta^{17}+\beta^{28}$로 표현되고, $\{\alpha_0,\alpha_1,\cdots,\alpha_6\}$는 NB이다. $0\le i\le 6$에 대해, $\alpha\alpha_i$의 계산은 표 1로부터 얻을 수 있다. 각 블록 $K$와 $K'$는 엔트리 $(s,t)(0\le s\le 3,\ 0\le t\le 6)$에 대해 각각 $\tau^s 2^t$와 $1+\tau^s 2^t$의 값을 가진다.

표 1. $GF(2^7)$상에서 GNB 타입 4를 이용한 $K_i$와 $K_i'$의 계산

| $K_0$ | $K_1$ | $K_2$ | $K_3$ | $K_4$ | $K_5$ | $K_6$ | $K_0'$ | $K_1'$ | $K_2'$ | $K_3'$ | $K_4'$ | $K_5'$ | $K_6'$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 8 | 16 | 3 | 6 | 2 | 3 | 5 | 9 | 16 | 4 | 7 |
| 12 | 24 | 19 | 9 | 18 | 7 | 14 | 13 | 25 | 20 | 10 | 19 | 8 | 15 |
| 28 | 27 | 25 | 21 | 13 | 26 | 23 | 0 | 28 | 26 | 22 | 14 | 27 | 24 |
| 17 | 5 | 10 | 20 | 11 | 22 | 15 | 18 | 6 | 11 | 21 | 12 | 23 | 16 |

표 1로부터 $\alpha\alpha_0=\alpha_1$이고, 나머지 부분은 식 (4)와 같다.

$$\begin{aligned}&\alpha\alpha_1=\alpha_0+\alpha_2+\alpha_5+\alpha_6,\\ &\alpha\alpha_2=\alpha_1+\alpha_3+\alpha_4+\alpha_5,\\ &\alpha\alpha_3=\alpha_2+\alpha_5,\\ &\alpha\alpha_4=\alpha_2+\alpha_5,\\ &\alpha\alpha_5=\alpha_0+\alpha_2+\alpha_5+\alpha_6,\\ &\alpha\alpha_6=\alpha_1+\alpha_3+\alpha_4+\alpha\end{aligned} \tag{4}$$

예를 들어, 블록 $K_2'$에 대한 $\alpha\alpha_2$는 다음과 같이 나타낼 수 있다. 블록 $K_2'$의 원소는 5, 20, 26, 11이며 $K_i$ 블록들의 $5\in K_1$, $20\in K_3$, $26\in K_5$, $11\in K_4$에서 찾을 수 있다. 따라서 $\alpha\alpha_2=\alpha_1+\alpha_3+\alpha_4+\alpha_5$이다. 식 (4)로부터 곱셈 행렬 $(\lambda_{ij})$은 식 (5)와 같다.

$$(\lambda_{ij})=\begin{pmatrix}0&1&0&0&0&0&0\\1&0&1&0&0&1&1\\0&1&0&1&1&1&0\\0&0&1&0&0&1&0\\0&0&1&0&0&0&1\\0&1&1&1&0&0&1\\0&1&0&0&1&1&1\end{pmatrix} \tag{5}$$

## 2.2 GNB를 이용한 $GF(2^m)$상의 비트-레벨 곱셈 알고리즘

$A=\sum_{i=0}^{m-1}a_i\alpha_i$, $B=\sum_{j=0}^{m-1}b_j\alpha_j$를 $GF(2^m)$상의 두 원소라 하면, 두 원소의 곱 $C=AB=\sum_{s=0}^{m-1}c_s\alpha_s$는 식 (6)과 같다.

$$\begin{aligned}C&=\sum_{i=0}^{m-1}a_i\alpha_i\sum_{j=0}^{m-1}b_j\alpha_j=\sum_{i,j}a_ib_j\alpha_i\alpha_j\\ &=\sum_{i,j}a_ib_j\sum_{s=0}^{m-1}\lambda_{ij}^{(s)}=\sum_{s=0}^{m-1}(\sum_{i,j}a_ib_j\lambda_{ij}^{(s)})\alpha_s\end{aligned} \tag{6}$$

또한 Kwon[15]의 보조 정리 1로부터 $C=\sum_{s=0}^{m-1}c_s\alpha_s$의 계수 $c_s$는

$$\begin{aligned}c_s&=\sum_{i,j}a_{i+s}b_{j+s}\lambda_{ij}^{(0)}=\sum_{i,j}a_{i+s}b_{j+s}\lambda_{ij}\\ &=\sum_{j=0}^{m-1}(\sum_{i=0}^{m-1}a_{i+s}\lambda_{ij})b_{j+s}\end{aligned} \tag{7}$$

와 같다. $m\times m$행렬 $X=(x_{st})$에 대해 $GF(2)$상의 원소 $x_{st}$, $0\le s,t\le m-1$을 식 (8)과 같이 정의하면,

$$x_{st}=(\sum_{i=0}^{m-1}a_{i+s}\lambda)b_{t+s} \tag{8}$$

$X$의 $t$번째 열벡터 $X_t$는 식 (9)와 같다.

$$X_t=(x_{0t},x_{1t},\cdots,x_{m-1,t})^T \tag{9}$$

여기서 $(x_{0t},x_{1t},\cdots,x_{m-1,t})^T$는 행벡터 $(x_{0t},x_{1t},\cdots,x_{m-1,t})$의 전치 행렬이다. 또한 $\sum_{t=0}^{m-1}x_{st}=c_s$이기 때문에 모든 열벡터 $X_t$, $t$=0,1,⋯,$m$-1의 합은 식 (10)과 같다.

$$(c_0,c_1,\cdots,c_{m-1})^T \tag{10}$$

여기서 $m$-1=2$v$라 하고 $Y=(y_{st})$를 $X$의 열벡터 치환에 의한 $m\times m$행렬이라 정의하면, $v$가 홀수일 때 $Y$는 식 (11)과 같이 정의되고,

$$\begin{aligned}Y=(&X_v,\cdots,X_3,X_1,X_{m-1},X_{m-3},\cdots,X_{m-v},X_{v-1},\cdots\\ &X_2,X_0,X_{m-2},\cdots,X_{m-v+1})\end{aligned} \tag{11}$$

$v$가 짝수일 때, $Y$는 식 (12)와 같이 정의된다.

$$\begin{aligned}Y=(&X_v,\cdots,X_2,X_0,X_{m-2},\cdots,X_{m-v},X_{v-1},\cdots,\\ &X_3,X_1,X_{m-1},X_{m-3},\cdots,X_{m-v+1})\end{aligned} \tag{12}$$

여기서, $Y_t=(y_{0t},y_{1t},\cdots,y_{m-1,t})^T$인 $Y$의 모든 열벡터 $Y_t$, $t$=0,1,⋯,$m$-1의 합은 $(c_0,c_1,\cdots,c_{m-1})^T$인 $X$의 모든 열

벡터 $X_t$, $t$=0,1,⋯,$m$-1의 합과 같다. 따라서 패러럴 입출력 곱셈기를 설계하기 위해 $Y$의 열벡터 합을 계산하는 대신 $Y$의 순환 쉬프트된 대각 벡터의 합을 계산한다. 즉, 행렬 $Y$의 표현에서 벡터 $X_t$와 $X_{m-t}$ 사이에 정확히 $t$-1개의 열이 존재한다. 또한 $X_t$의 $s$ 번째 원소와 $X_{m-t}$의 $s+t$번째 원소는 그들의 가수(summand)에서 동일한 $a_{is}$를 가진다. 즉, 식 (8)로부터 식 (13)을 얻을 수 있다.

$$\begin{aligned} x_{s+t,m-t} &= \Big(\sum_{i=0}^{m-1} a_{i+s+t}\lambda_{i,m-t}\Big)b_s \\ &= \Big(\sum_{i=0}^{m-1} a_{i+s}\lambda_{i-t,m-t}\Big)b_s \\ &= \Big(\sum_{i=0}^{m-1} a_{i+s}\lambda\Big)b_s \end{aligned} \tag{13}$$

여기서 $x_{st}$와 $x_{s+t,m-t}$는 같은 항 $\sum_{i=0}^{m-1} a_{i+s}\lambda$를 가지며 이는 $AB$의 계산에 있어 사용되는 XOR 게이트의 개수를 줄인다. 지금까지 설명한 내용을 바탕으로 [알고리즘 1]를 얻을 수 있다.

---

[알고리즘 1] GNB를 이용한 $GF(2^m)$상의 비트-레벨 곱셈 알고리즘[6]

---

Input : $A, B \in GF(2^m)$

Output : $D$, $D_i = c_i$ for all $0 \le i \le m-1$,

where $AB = \sum_{i=0}^{m-1} c_i \alpha_i$

Initial : $A \leftarrow (a_0, a_1, \cdots, a_{m-1})$

$B \leftarrow (b_0, b_1, \cdots, b_{m-1})$

$D \leftarrow (D_0, D_1, \cdots, D_{m-1}) \leftarrow (0,0,\cdots,0)$

1. for $0 \le t \le m-1$
2. for $0 \le s \le m-1$
3. $D_{s+t+1} \leftarrow y_{s,s+t} + D_{s+t}$
4. end for
5. end for
6. return $D$

---

## Ⅲ. GNB를 이용한 $GF(2^m)$상의 새로운 워드-레벨 곱셈기

### 3.1 GNB를 이용한 $GF(2^m)$상의 워드-레벨 곱셈 알고리즘

[알고리즘 1]로부터 비트-시리얼 또는 비트-패러럴 곱셈기를 쉽게 유도 할 수 있다. 그러나 비트-시리얼 구조는 속도가 너무 느리고 비트-패러럴 구조는 큰 $m$(최소 163)을 요구하는 ECC와 같은 응용에는 적합하지 않다. 이러한 이유로 많은 타원곡선 암호 프로세서들은 워드-레벨 곱셈기를 채택하였다[5],[10]. 워드-레벨 구조는 데이터 크기가 $m$-비트이고 워드의 크기가 $w$-비트이면 워드의 개수는 $L = \lceil m/w \rceil$이 되고 $L$클럭 사이클마다 결과를 출력한다. [알고리즘 1]로부터 [알고리즘 2]와 같은 $GF(2^m)$상의 워드-레벨 곱셈 알고리즘을 얻을 수 있다.

---

[알고리즘 2] GNB를 이용한 $GF(2^m)$상의 워드-레벨 곱셈 알고리즘

---

Input : $A$, $B \in GF(2^m)$

Output : $D = (D_0, D_1, \cdots, D_{m-1})$,

$D_S = c_s$ for all $0 \le s \le m-1$,

where $AB = \sum_{s=0}^{m-1} c_s \alpha_s$

Initial : $A \leftarrow (a_0, a_1, \cdots, a_{m-1})$

$B \leftarrow (b_0, b_1, \cdots, b_{m-1})$

$D \leftarrow (D_0, D_1, \cdots, D_{m-1}) \leftarrow (0,0,\cdots,0)$

1. for $0 \le t \le L-2$
2. for $0 \le s \le m-1$
3. $D_{s+(t+1)w-\gamma} \leftarrow y_{s,s+tw} + y_{s,s+tw+1} + \cdots + y_{s,s+tw+(w-1)} + D_{s+tw-\gamma}$
4. end for
5. end for
6. $t = L-1$
7. for $0 \le s \le m-1$
8. $D_{s+(t+1)w-\gamma} \leftarrow y_{s,s+tw} + y_{s,s+tw+1} + \cdots + y_{s,s+tw+(w-1)-\gamma} + D_{s+tw-\gamma}$
9. end for
10. return $D$

---

$\gamma = L \cdot w - m$, $L = \lceil m/w \rceil$

[알고리즘 2]에서 $0 \le t \le L-2$일 때, 모든 $0 \le s \le m-1$에 대한 $w+1$개의 항($y_{s,s+tw} + y_{s,s+tw+1} + \cdots + y_{s,s+tw+(w-1)} + D_{s+tw-\gamma}$)은 동시에 계산된다. 하지만 $t=L-1$일 때는 $\gamma$개의 블록이 $t=0$일 때의 원소와 중복되므로 모든 $0 \le s \le m-1$에 대한 $w-\gamma+1$개의 항($y_{s,s+tw} + y_{s,s+tw+1} + \cdots + y_{s,s+tw+(w-1)-\gamma} + D_{s+tw-\gamma}$)이 동시에 계산된다. 즉, 고정된 $s$에 대한 마지막 출력값 $D_s$는 다음 식과 같이 연속적으로 계산된다.

$$\begin{aligned} D_s &= \overbrace{y_{s,s} + y_{s,s+1} + \cdots + y_{s,s+(\omega-1)}}^{D_{s+\omega}} + y_{s,s+\omega} + \\ &\quad \underbrace{y_{s,s+\omega+1} + \cdots + y_{s,s+2\omega-1}}_{D_{s+2\omega}} \\ &\quad + y_{s,s+2\omega} + \cdots + y_{s,s+m-1} = \sum_{i=0}^{m-1} y_{s,s+i} = c_s \end{aligned} \tag{14}$$

NIST에 의해 권고된 $GF(2^m)$상의 필드 크기가 소수이기 때문에 항상 $\Upsilon \neq 0$이며 연산의 마지막 반복에서 중복되는 계산을 피하기 위해 $m \cdot \Upsilon$ 항이 구분되어 수행되어야한다. 여기서 $m \cdot \Upsilon$ 항은 [알고리즘 2]의 $L$-1번째 단계에서 계산되기 때문에 규칙적인 하드웨어 구조 설계를 어렵게 한다. 즉, 하드웨어 구조는 [알고리즘 2]의 단계 3, 8을 수행하기 위해 각각 다르게 설계되어야 한다. 본 논문에서는 이러한 문제를 해결하기 위해 $L$길이의 컨트롤 신호(111,⋯,10)를 사용한다. 보다 자세한 설명은 2절에 기술한다.

### 3.2 GNB 타입 4를 이용한 $GF(2^m)$상의 워드-레벨 곱셈기 설계

$$\begin{aligned}
c_0 &= \underline{(a_2+a_5)b_3+a_{0256}b_1}+a_{1456}b_6+(a_2+a_6)b_4+a_{1345}b_2+a_1b_0+a_{1236}b_5 \\
c_1 &= (a_3+a_6)b_4+\underline{a_{1360}b_2+a_{2560}b_0}+(a_3+a_0)b_5+a_{2456}b_3+a_2b_1+a_{2340}b_6 \\
c_2 &= (a_4+a_0)b_5+a_{2401}b_3+\underline{a_{3601}b_1+(a_4+a_1)b_6}+a_{3560}b_4+a_3b_2+a_{3451}b_0 \\
c_3 &= (a_5+a_1)b_6+a_{3512}b_4+a_{4012}b_2+\underline{(a_5+a_2)b_0+a_{4601}b_5}+a_4b_3+a_{4562}b_1 \\
c_4 &= (a_6+a_2)b_0+a_{4623}b_5+a_{5123}b_3+(a_6+a_3)b_1+\underline{a_{5012}b_6+a_5b_4}+a_{5603}b_2 \\
c_5 &= (a_0+a_3)b_1+a_{5034}b_6+a_{6234}b_4+(a_0+a_4)b_2+a_{6123}b_0+\underline{a_6b_5+a_{6014}b_3} \\
c_6 &= \underline{(a_1+a_4)b_2}+a_{6145}b_0+a_{0345}b_5+(a_1+a_5)b_3+a_{0234}b_1+a_0b_6+\underline{a_{0125}b_4}
\end{aligned} \quad (15)$$

예제로 $w=2$인 $GF(2^7)$상의 워드-레벨 곱셈기를 설계한다. 식 (5)의 곱셈 행렬과 식 (7), (8), (11), (12)로부터 곱셈 결과 $C=AB=\sum_{i=0}^{6} c_i\alpha_i$는 식 (15)와 같이 얻을 수 있다. 식 (15)의 밑줄은 $w=2$, $a_{ijkl}$는 $a_i+a_j+a_k+a_l$을 의미한다. 그림 1은 $w=2$에 대하여 $GF(2^7)$상에서 GNB 타입 4를 이용한 $C=AB$에 대응되는 쉬프트 레지스터 회로이고, 그림 2는 그림 1의 $f_j$ 블록 구조를 나타낸다. 밑줄 친 두 개의 원소를 XOR 연산하여 레지스터에 저장하며 밑줄 친 원소의 첫 번째 항은 $a_i$의 공통된 항을 가지는 주대각 원소를 계산하는 $f_0$ 블록이고, 두 번째 항은 주대각 원소의 벡터 $a_i$, $b_i$를 한 번 순환 쉬프트하여 계산하는 $f_1$ 블록이다. 워드-레벨 구조에서는 한 클럭 사이클마다 $w=2$개의 원소를 연산하므로 $A$, $B$, $R$ 레지스터는 $w$ 크기만큼 순환 쉬프트하며 $L=4$ 클럭 사이클 후에 모든 연산이 끝나고 결과가 출력된다. $m$이 홀수이기 때문에 두 개의 원소씩 처리하면 마지막 클럭 사이클의 연산에 중복되는 원소가 나타나며 이를 회피해야 한다. 워드의 크기 $w$에 따라 중복되는 원소의 개수 $w$-$\Upsilon$도 다르게 나타나며 중복되는 원소의 개수만큼 마지막 클럭 사이클에서 연산을 회피해야 한다. 그림 1의 곱셈기는 (1,1,1,0)의 제어 신호를 사용하여 마지막 제어 신호에서 중복되는 값을 회피한다. 또한 $m/w$가 정수가 아니기 때문에 곱셈 결과 레지스터 $R_i$는 정확한 $c_i$를 가지지 않는다. 따라서 정확한 출력을 위해 $R_i$는 $\Upsilon$만큼의 순환 쉬프트가 필요하다.

## Ⅳ. FPGA 구현 및 성능분석

본 논문에서 제안한 $GF(2^m)$상의 워드-레벨 곱셈기의 FPGA 구현 및 기능 검증을 위해 VHDL로 회로를 기술하였고, Xilinx사의 ISE 6.3i를 이용하여 회로를 합성한 후 Mentor Graphics사의 Model Sim을 이용하여 그 기능을 검증하였다. 정확한 기능 검증을 한 후 리버트론사의 SoC(System-on-Chip) 테스트 보드를 이용하여 FPGA에 워드-레벨 곱셈기를 구현하였다. 테스트 보드는 ARM7TDMI 마이크로프로세서와 Xilinx XC2V1000 FPGA칩을 탑재하고 있으며, 최대 50MHz의 클럭 주파수를 지원한다.

표 2에는 기존에 제안된 워드-레벨 곱셈기와 성능을 비교하였고 표 3에 제안된 워드-레벨 곱셈기의

표 2. ONB 타입 II에 대한 워드-레벨 곱셈기 비교

| Multipliers | Critical Delay | #AND | #XOR | #FF | Output |
|---|---|---|---|---|---|
| 제안된 곱셈기 | $2T_A+(1+\lceil \log_2(w+1) \rceil)T_X$ | $wm+m$ | $w(3m-1)/2$ | $3m$ | parallel |
| DLMO[13] | $T_A+(1+\lceil \log_2(2m-1) \rceil)T_X$ | $w(2m-1)$ | $w(2m-2)$ | $2m$ | serial |
| IMO[14] | $T_A+(1+\lceil \log_2 m \rceil)T_X$ | $wm$ | $w(2m-2)$ | $2m$ | serial |
| AEDS[11] | $T_A+(\lceil \log_2(2m-1) \rceil)T_X$ | $w(m-0.5w+0.5)$ | $w(3m-d-2)$ | $2m$ | serial |
| XEDS[11] | $T_A+(\lceil \log_2(2m-1) \rceil)T_X$ | $w(2m-w)$ | $w(2m-0.5w-1.5)$ | $2m$ | serial |
| $w$-SMPOI[12] | $2T_A+(3+\lceil \log_2(w-1) \rceil)T_X$ | $w(m+1)/2+m$ | $w(2m-1)$ | $3m$ | parallel |
| $w$-SMPOII[12] | $2T_A+(3+\lceil \log_2(w-1) \rceil)T_X$ | $wm+m$ | $w(3m-1)/2$ | $3m$ | parallel |

표 3. GNB 타입 $k$에 대한 그림 1 곱셈기의 성능 분석

| Critical Delay | #AND | #XOR | #FF | Output |
|---|---|---|---|---|
| $\leq 2T_A+(\lceil \log_2 k \rceil+\lceil \log_2(w+1) \rceil)T_X$ | $wm+\Upsilon m$ | $\leq wm+w(m-1)(k-1)/2$ | $3m$ | parallel |

$\Upsilon = L \cdot w - m$, $L=\lceil m/w \rceil$

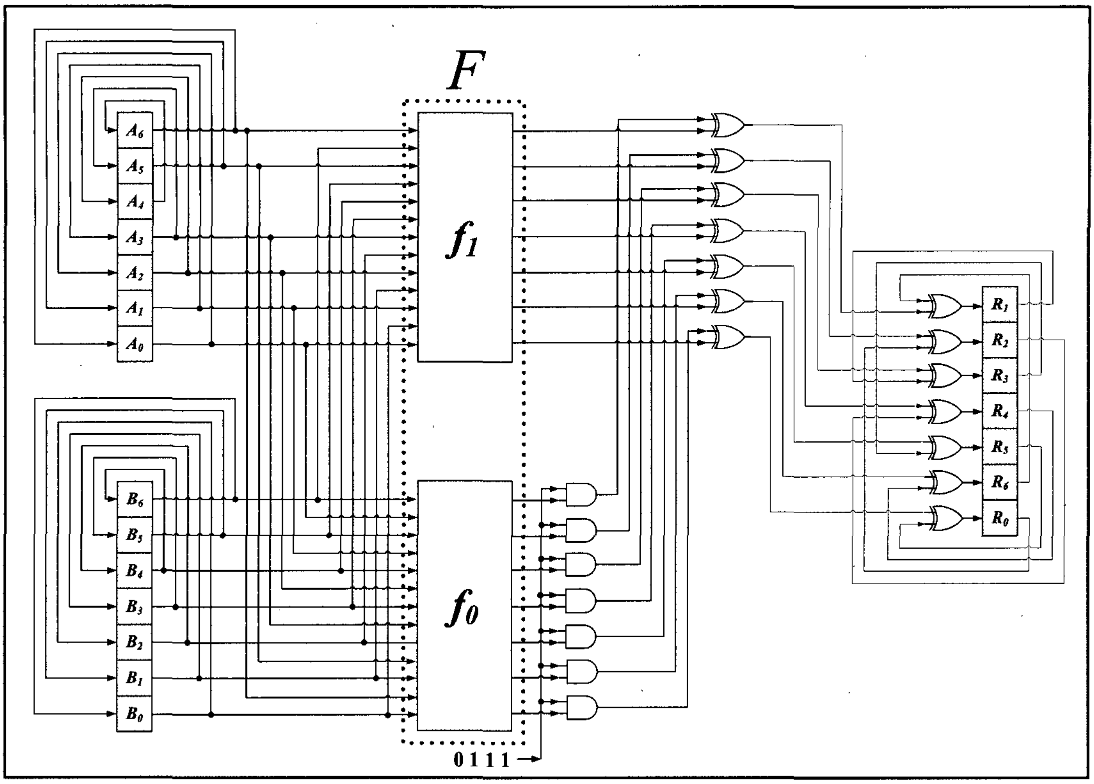



그림 1. $m$=7, $w$=2에 대한 $GF(2^m)$상의 GNB 타입 4 곱셈기

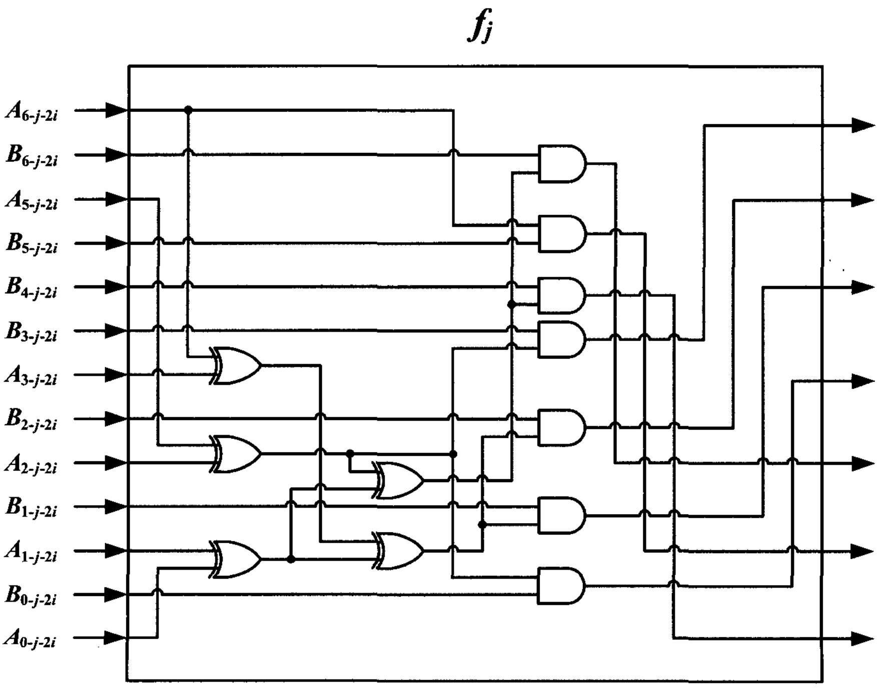



그림 2. 그림 1의 $f_j$

일반적인 GNB 타입에 대한 성능을 요약하였다. 표 4에는 ECC를 위하여 NIST에서 권고하는 $GF(2^m)$ 필드 크기 중에서 가장 작은 $m=163$을 선택하고 다양한 $w$에 대한 FPGA 구현결과를 요약하였다. 표 2에서 기존의 곱셈기는 최적 정규기저(Optimal Normal Basis: ONB)를 사용하기 때문에 본 논문에서 제안된 곱셈기의 GNB 타입 2와 비교하였다. 표 2에 나타나듯이 본 논문에서 제안된 곱셈기가 시리얼 출력 형태를 가지는 곱셈기들 보다 총 게이트수가 적으며 모든 곱셈기와 비교했을 때 가장 낮은 처리기 지연 시간을 보인다. 제안된 워드-레벨 곱셈기는 이론적으로 타입 $k$에 대해 처리 지연 시간이 $2T_A+(\lceil \log_2 k \rceil + \lceil \log_2 w+1 \rceil)T_X$이고 $wm+\gamma m$개의 AND 게이트, $wm+w(m-1)(k-1)/2$개의 XOR 게이트, $3m$개의 Flip-Flop(FF)이 필요하다. 여기서 $T_A$는 2-입력 AND 게이트 지연시간이고 $T_X$는 2-입력 XOR 게이트의 지연시간이다.

표 4. $m=163$에 대한 그림 1 곱셈기의 구현 결과

| w | LUT 개수 | FF 개수 | 최대동작 주파수(MHz) |
|---|---|---|---|
| 8 | 2113 | 489 | 197.873MHz |
| 16 | 5127 | 489 | 171.807MHz |
| 32 | 8927 | 489 | 149.058MHz |
| 64 | 18901 | 489 | 142.248MHz |

## V. 결 론

본 논문에서는 Kwon 등[6]에 의해 제안된 $GF(2^m)$상의 비트-레벨 곱셈 알고리즘을 기반으로 $GF(2^m)$상의 워드-레벨 곱셈 알고리즘을 유도하였으며, 이를 바탕으로 GNB를 이용한 $GF(2^m)$상의 워드-레벨 곱셈기를 설계하였다. 또한 ECC를 위해 $m$=163을 선택하고 다양한 $w$에 대해 FPGA를 구현한 후 성능을 측정하였다. 본 논문에서 제안된 곱셈기와 기존에 제안된 곱셈기의 성능을 비교 분석한 결과, 가장 낮은 최대 처리기 지연시간을 보였다. 더욱이 기존의 곱셈기는 선택된 원시 기약 다항식에 따라 서로 다른 하드웨어 구조를 가지는데 반해 본 논문에서 제안된 곱셈기는 동일한 GNB 타입만 가지면 동일한 형태의 하드웨어 구조를 얻을 수 있다. 따라서 본 논문에서 제안된 워드-레벨 곱셈기는 ECC의 곱셈기로 매우 적합하다 할 수 있다.

## 참 고 문 헌


[1] IEEE 1363, *Standard Specifications for Publickey Cryptography*, 2000.

[2] NIST, Recommended elliptic curves for federal government use, May 1999. http://csrc.nist.gov/encryption.

[3] A. Reyhani-Masoleh and M.A. Hasan, "A New Construction of Massey-Omura Parallel Multipliers over $GF(2^m)$," *IEEE Transactions on Computers*, Vol. 51, No. 5, pp. 511-520, May. 2002.

[4] M.C. Rosner, "Elliptic Curve Cryptosystems on Reconfigurable Hardware," *MA thesis, Worcester Polytechnic Institute*, 1998.

[5] G. Orlando and C. Parr, "A High Performance Reconfigurable Elliptic Curve Processor for $GF(2^m)$," *CHES 2000*, LNCS 1965, 2000.

[6] S. Kwon, K. Gaj, C.H. Kim, and C.P. Hong, "Efficient Linear Array for Multiplication in $GF(2^m)$ Using a Normal Basis for Elliptic Curve Cryptography," *CHES 2004*, LNCS 3156, pp. 76-91, 2004.

[7] J.R. Goodman, Energy Scalable Reconfigurable Cryptographic Hardware for Portable Applications, PhD thesis, MIT, 2000.

[8] J.H. Guo and C.L. Wang, "Digit-Serial Systolic Multiplier for Finite Field $GF(2^m)$," *IEE Proc. Comput. Digit. Tech.*, vol. 145, no 2, pp. 143-148, Mar. 1999.

[9] C.H. Kim, S.D. Han and C.P. Hong, "An Efficient Digit-Serial Systolic Multiplier for Finite Field $GF(2^m)$," *Proc. on 14th Annual IEEE International Conference of ASIC/SOC*, pp. 361-365, 2001.

[10] N. Gura, S.C. Shantz, H.E. Sumit Gupta, V. Gupta, D. Finchelstein, E. Goupy, and D. Stebila, "An End-to-End Systems Approach to Elliptic Curve Cryptography," *CHES '02*, LNCS 2523, pp. 349-365, 2002.

[11] A. Reyhani-Masoleh and M.A. Hasan, "Efficient Digit-Serial Normal Basis Multipliers over $GF(2^m)$," *ACM Trans. Embedded Computing Systems (TECS)*, special issue on embedded systems and security, vol. 3, no. 3, pp. 575-592, Aug. 2004.

[12] A. Reyhani-Masoleh and M.A. Hasan, "Low Complexity Word-Level Sequential Normal

Basis Multipliers," *16th IEEE Transactions on Computers*, vol. 54, No 2, pp. 98-110, 2005.
〔13〕 J. L. Massey and J.K. Omura, "Computational method and apparatus for finite field arithmetic," *US Patent No. 4587627*, 1986.
〔14〕 L. Gao and G.E. Sobelman, "Improved VLSI Designs for Multiplication and Inversion in $GF(2^M)$ over Normal Bases," *Proc. 13th Ann. IEEE Int'l ASIC/SOC Conf.*, pp.97-101, 2000.

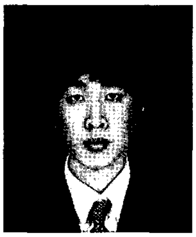
**김 창 훈** (Chang Hoon Kim) 정회원

2001년 2월 대구대학교 컴퓨터 정보공학부, 학사
2003년 2월 대구대학교 컴퓨터 정보공학과, 석사
2006년 8월 대구대학교 컴퓨터 정보공학과, 박사
2006년 9월~현재 대구대학교 정보통신공학과 BK21, 연구교수
<관심분야> 암호 시스템, Embedded System, RFID/USN 보안

**권 윤 기** (Yun Ki Kwon) 준회원

2001년 2월 대진대학교 수학과, 학사
2003년 8월 성균관대학교 수학과, 석사
2003년 9월~현재 성균관대학교 수학과, 박사과정
<관심분야> 공개키 암호시스템, 암호시스템 구현, 타원곡선 암호시스템, Pairing 기반 암호시스템

**김 태 호** (Tae Ho Kim) 준회원



2006년 2월 대구대학교 컴퓨터 IT공학부, 학사
2006년 3월~현재 대구대학교 정보통신공학과, 석사과정
<관심분야> 암호 시스템, Embedded System, RFID/USN 보안

**권 순 학** (Soonhak Kwon) 정회원

1990년 2월 KAIST 수학과, 학사
1992년 2월 서울대학교 수학과, 석사
1997년 5월 Johns Hopkins University, 박사
1998년 3월~현재 성균관대학교 수학과, 부교수
<관심분야> 정수론, 암호론, Cryptographic Hardware

**홍 춘 표** (Chun Pyo Hong) 정회원

1978년 2월 경북대학교 전자공학과, 학사
1986년 12월 Georgia Institute of Technology ECE, 석사
1991년 12월 Georgia Institute of Technology ECE, 박사
1994년 9월~현재 대구대학교 정보통신공학부, 교수
<관심분야> DSP 하드웨어 및 소프트웨어, 컴퓨터 구조, VLSI 신호처리, Embedded System